

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) An apparatus for implementing a Floating-Point related application, comprising:

a tool that includes:

a receiver for receiving a list of floating-point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest in respect ~~of~~ to at least one ~~FP~~ floating-point instruction;

a parser for parsing the floating-point commands;

a processor configured to process at least the parsed commands for realizing the Floating-Point related application on the basis of said events.

2. (original) The apparatus of Claim 1, wherein said language further defining regrouping of the events into at least one coverage model; and wherein said processor is configured to process the parsed commands for realizing the floating-point related application on the basis of said events and said at least one coverage model.

3. (original) The apparatus according to Claim 1, wherein said application is an evaluation of coverage of tests being run on a design.

4. (original) The apparatus according to Claim 1, wherein said processor is configured to generate a sequence of test vectors for verification of Floating-Point module operation; the test vectors meet the constraints of said events.

5. (original) The apparatus according to Claim 4, wherein said verification includes verifying if the Floating-Point operation complies with IEEE standard for Floating-Point.

6. (currently amended) For use with the Floating-Point module of Claim 1, a computer language; the language defining Floating-Point events of interest in respect of to at least one FP floating-point instruction.

7. (original) The computer language of Claim 6, further defining regrouping of the events into at least one coverage model.

8. (currently amended) An apparatus for implementing a Floating-Point related application, comprising:

a tool that includes:

a receiver for receiving a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of to at least one FP floating-point instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP floating-point instruction, and (ii) result operand of the FP floating-point instruction;

a parser for parsing the floating point commands;

a processor for processing at least the parsed commands for realizing the Floating-Point related application at least on the basis of said events and said at least one coverage model.

9. (original) The apparatus according to Claim 8, wherein said application is an evaluation of coverage of tests being run on a design.

10. (original) The apparatus according to Claim 8, wherein said processor is configured to generate a sequence of test vectors for verification of Floating-Point module operation; the test vectors meet the constraints of said events and the at least one coverage model.

11. (original) The apparatus according to Claim 8, wherein said verification includes verifying if the Floating-Point operation complies with the IEEE standard for Floating-Point.

12. (currently amended) For use with the Floating-Point module of Claim 8, a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect ~~of~~ to at least one FP floating-point instruction, the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP floating-point instruction, and (ii) result operand of the FP floating-point instruction.

13. (currently amended) An apparatus for implementing a Floating-Point related application, comprising:
a tool that includes:
a receiver for receiving a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect ~~of~~ to at least one FP floating-point instruction; the coverage model having the form of a sequence of

Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP floating-point instruction, and (ii) result operand of the FP floating-point instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

a parser for parsing the floating point commands;
a processor for processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-Point related application.

14. (original) The apparatus according to Claim 13, wherein said application is an evaluation of coverage of tests being run on a design.

15. (original) The apparatus according to Claim 13, wherein said processor is configured to generate a sequence of test vectors for verification of Floating-Point module operation; the test vectors meet the constraints of said events and at least one coverage model.

16. (original) The apparatus according to Claim 13, wherein said verification includes verifying if the Floating-Point operation complies with the IEEE standard for Floating-Point.

17 (currently amended) For use with the Floating-Point module of Claim 13, a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of to at least one FP floating-point instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on:

(i) at least one intermediate result operand of the FP floating-point instruction, and (ii) result operand of the FP floating-point instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers.

18. (currently amended) The apparatus according to Claim 1, wherein said list of commands includes: Range of FP floating-point numbers ; Mask on bits of FP floating-point number; Set or Reset Number of Bits in ~~an FP~~ a floating-point number; Set or Reset Continuous-Bit-Long in ~~an FP~~ a floating-point number; Relative Values of at least two FP floating-point numbers, and logical operations among said commands.

19. (currently amended) The apparatus according to Claim 6, wherein said list of commands includes: Range of FP floating-point numbers; Mask on bits of FP floating-point number; Set or Reset Number of Bits in ~~an FP~~ a floating-point

number; Set or Reset Continuous-Bit-Long in ~~an FP~~ a floating-point number; Relative Values of at least two FP floating-point numbers, and logical operations among said commands.

20. (currently amended) The apparatus according to Claim 7, wherein said list of commands includes: Range of FP floating-point numbers; Mask on bits of FP a floating-point number; Set or Reset Number of Bits in ~~an FP~~ a floating-point number; Set or Reset Continuous-Bit-Long in ~~an FP~~ a floating-point number; Relative Values of at least two FP floating-point numbers, and logical operations among said commands.

21. (currently amended) The apparatus according to Claim 8, wherein said list of commands includes: Range of FP floating-point numbers ; Mask on bits of FP a floating-point number; Set or Reset Number of Bits in ~~an FP~~ a floating-point number; Set or Reset Continuous-Bit-Long in ~~an FP~~ a floating-point number; Relative Values of at least two FP floating-point numbers, and logical operations among said commands.

22. (currently amended) The apparatus according to Claim 12, wherein said list of commands includes: Range of FP floating-point numbers ; Mask on bits of FP a floating-point number; Set or Reset Number of Bits in ~~an FP~~ a floating point number; Set or Reset Continuous-Bit-Long in ~~an FP~~ a floating-

point number; Relative Values of at least two FP floating-point numbers, and logical operations among said commands.

23. (currently amended) The apparatus according to Claim 13, wherein said list of commands includes: Range of FP floating-point numbers ; Mask on bits of FP a floating-point number; Set or Reset Number of Bits in ~~an~~ FP a floating-point number; Set or Reset Continuous-Bit-Long in ~~an~~ FP a floating-point number; Relative Values of at least two FP floating-point numbers, and logical operations among said commands.

24. (currently amended) The apparatus according to Claim 17, wherein said list of commands includes: Range of FP floating-point numbers ; Mask on bits of FP a floating-point number; Set or Reset Number of Bits in ~~an~~ FP a floating-point number; Set or Reset Continuous-Bit-Long in ~~an~~ FP a floating-point number; Relative Values of at least two FP floating-point numbers, and logical operations among said commands.

25. (original) The apparatus according to Claim 8, wherein said constraints are further applied to attributes of Machine State.

26. (original) The apparatus according to Claim 13, wherein said constraints are further applied to attributes of Machine State.

27. (currently amended) A method for implementing a Floating-Point related application that includes the steps of :

(a) receiving a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest in respect ~~of~~ to at least one FP floating-point instruction;

(b) parsing the floating point commands; and

(c) processing at least the parsed commands for realizing the Floating-Point related application on the basis of said events.

28. (currently amended) A method for implementing a Floating-Point related application in a computer system that includes the steps of:

(a) receiving a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect ~~of~~ to at least one FP floating-point instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP floating-point instruction, and (ii) result operand of the FP floating-point instruction;

(b) parsing the floating point commands; and

(c) processing at least the parsed commands for realizing the Floating-Point related application at least on the basis of said events and said at least one coverage model.

29. (currently amended A method for implementing a Floating-Point related application in a computer system, that includes the step of:

(a) receiving a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of to at least one FP floating-point instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP floating-point instruction, and (ii) result operand of the FP floating-point instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

(b) parsing the floating point commands; and

(c) processing at least the parsed commands for realizing at least on the basis of said events and said at

least one coverage model the Floating-Point related application.

30. (currently amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a Floating-Point related application that includes the steps of :

(a) receiving a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest in respect of to at least one FP floating-point instruction;

(b) parsing the floating point commands; and

(c) processing at least the parsed commands for realizing the floating-point related application on the basis of said events.

31. (currently amended) A computer program product comprising a computer useable medium having computer readable program code embodied therein for causing the computer to implement a Floating-Point related application, comprising:

computer readable program code for causing the computer to receive a list of floating point commands in a formal computer language; the language having set based

constraints that facilitate defining Floating-Point events of interest in respect of to at least one FP floating-point instruction;

computer readable program code for causing the computer to parse the floating point commands; and

computer readable program code for causing the computer to process at least the parsed commands for realizing the Floating-Point related application on the basis of said events.

32. (currently amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a Floating-Point related application, that includes the steps of:

(a) receiving a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of to at least one FP floating-point instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP floating-point instruction, and (ii) result operand of the FP floating-point instruction; each one of said constraints is expressed as at

least one set each of which defining allowable Floating-Point numbers;

(b) parsing the floating point commands; and

(c) processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-Point related application.

33. (currently amended) A computer program product comprising a computer useable medium having computer readable program code embodied therein for causing the computer to implement a Floating-Point related application, comprising:

computer readable program code for causing the computer to receive a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect ~~of~~ to at least one ~~FP floating-point~~ instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the ~~FP floating-point~~ instruction, and (ii) result operand of the ~~FP floating-point~~ instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

computer readable program code for causing the computer to parse the floating point commands; and
computer readable program code for causing the computer to process at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-Point related application.

34. (currently amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a Floating-Point related application, that includes the steps of:

(a) receiving a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect ~~of~~ to at least one ~~FP~~ floating-point instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the ~~FP~~ floating-point instruction, and (ii) result operand of the ~~FP~~ floating-point instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

(b) parsing the floating point commands; and

(c) processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-Point related application.

35. (currently amended) A computer program product comprising a computer useable medium having computer readable program code embodied therein for causing the computer to implement a Floating-Point related application, comprising:

computer readable program code for causing the computer to receive a list of floating point commands in a formal computer language; the language having set based constraints that facilitate defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of to at least one FP floating-point instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP floating-point instruction, and (ii) result operand of the FP floating-point instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

computer readable program code for causing the computer to parse the floating point commands; and

computer readable program code for causing the

Appln. No. 09/927,457
Amd. dated September 22, 2005
Reply to Office Action of May 23, 2005

computer to process at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-Point related application.